

# Nintendo GameCube™ Video Guidelines

Version 1.5

## Table of Contents

<b>1. Introduction .....</b>	<b>4</b>
<b>2. The Effective Screen Frame.....</b>	<b>4</b>
<b>3. Safe Frames.....</b>	<b>5</b>
<b>4. Pixel Ratios .....</b>	<b>5</b>
4.1 Displaying on a normal size (4:3) TV.....	6
4.2 Displaying on a wide-screen (16:9) TV.....	6
<b>5. Black Bands on Both Edges of the TV Screen .....</b>	<b>7</b>
<b>6. MPAL .....</b>	<b>8</b>
<b>7. PAL .....</b>	<b>9</b>
7.1 Supporting Development Hardware .....	9
7.2 Handling EU RGB60 Mode .....	9
<b>8. Handling YUV Data .....</b>	<b>9</b>
<b>9. Problems with Screen Noise After IPL Ends .....</b>	<b>10</b>
<b>10. About Progressive Mode .....</b>	<b>10</b>
10.1 Switching between Interlaced Mode and Progressive Mode.....	10
10.2 Requirements when Using the Progressive Mode.....	11
10.3 Recommendations when Using Progressive Mode .....	12
10.3.1 Time to complete the switch.....	12
10.3.2 Processing Hardware Resets during Progressive Output.....	12
10.3.3 Other considerations for Progressive Mode .....	12
10.4 Interlaced/Progressive Mode Switching Flowchart .....	13
<b>11. Switching the Render Mode.....</b>	<b>14</b>
<b>12. About Televisions that Support Progressive Mode .....</b>	<b>14</b>
<b>13. Other Items .....</b>	<b>15</b>

## Table of Contents (continued)

### List of Figures

Figure 1: Effective screen frame (when IPL's "screen horizontal position" is 0).....	4
Figure 2: Safe frame examples (effective screen frame = 640 x 480).....	5
Figure 3: Pixel ratio when scaling is not used (for a 4:3 TV).....	6
Figure 4: Pixel ratio when scaling is not used (for a 16:9 TV).....	6
Figure 5: Black bands on both edges of the screen.....	7
Figure 6: Correspondence between TV screen and YUV data in the XFB .....	9
Figure 7: Flowchart Illustrating Switching between Interlaced and Progressive Modes .....	13

### List of Tables

Table 1: Example of screen width and safety frames taking into account the flicker of the black bands on both edges of the screen.....	8
Table 2: Market Survey of Differences between Televisions that Support Progressive Mode .....	12
Table 3: Sample of messages displayed on screen during switch between Interlaced and Progressive Modes.....	13

### List of Code

Code 1: Switching the Render Mode between NTSC Interlace and NTSC Non-interlace .....	14
---	----

### Revision History

Version	Revision Date	Description of Revisions
1.5	2/20/02	- Added Chapter 5 "Black Bands on Both Edges of the TV Screen" - Modified part of paragraph 10.1 for clarification
1.48	01/30/02	- Simplified the ratios in paragraph 4.1 and 4.2 - Added paragraphs 6.1 and 6.2 - Added sentence regarding progressive mode to end of paragraph 9.2
1.47	01/23/02	- Revised effective screen display height and description of non-display region in Chapter 2 - Revised MPAL description in Chapter 5 - Revised PAL description in Chapter 6 - Revised flow chart for interlaced/progressive mode switching in 9.4 - Added table of sample messages for interlaced/progressive switching to 9.4
1.44	10/23/01	- Revised description of Progressive Mode and its flow chart, Chapter 9 - Revised description contained in Chapter 11
1.42	9/12/01	- Revised paragraph 9.2 and Chapter 11
1.41	8/31/01	- Added Progressive Mode Guidelines (Chapter 9) - Added Figure 6 - Added Chapter 11 - Revised Chapters 10 and 12
1.12	8/3/01	- Revised Chapters 3 through 7, and 10
1.1	7/13/01	- Released by NOA

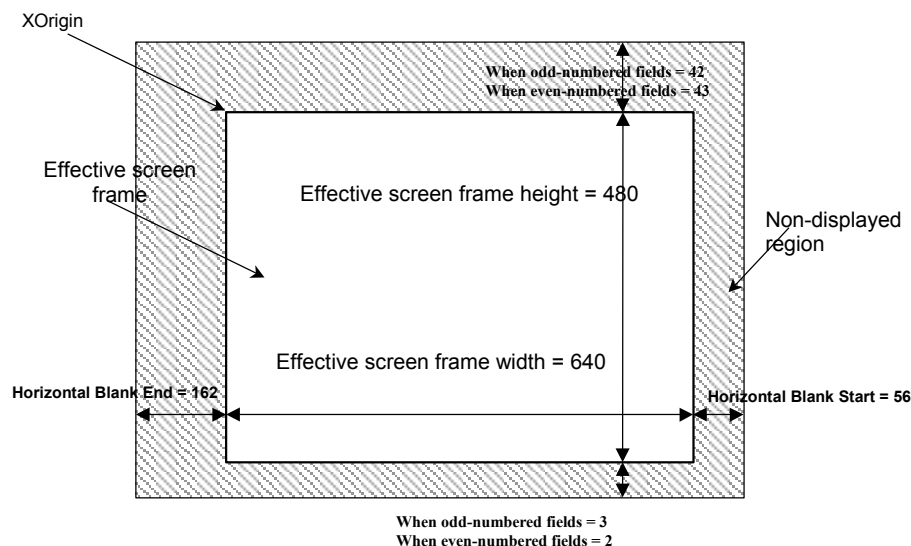
## 1. Introduction

These guidelines cover special warnings that apply when the video output from the Nintendo GameCube™ is displayed on a television set, together with information not handled in the Video Interface Library documentation.

Due to the large number of commercially available television sets, it is impossible to guarantee complete compatibility with all of them.

## 2. The Effective Screen Frame

The effective screen frame is the effective region to which the TV's image signals are displayed. For an NTSC interlace TV with an effective screen frame of 640 x 480, the Nintendo GameCube™ positions the effective screen frame as shown in Figure 1. The horizontal positioning of the screen for the Nintendo GameCube™ Initial Program Loader (IPL) can be altered. This involves scrolling the position by changing the Horizontal Blank End and Horizontal Blank Start values shown in Figure 1. For example, to scroll 1 pixel to the left, Horizontal Blank End would be set to 161 and Horizontal Blank Start would be set to 57.



**Figure 1: Effective screen frame (when IPL's "screen horizontal position" is 0)**

This frame can be made larger than the actual region displayed on the TV: the maximum is 720 x 480 for NTSC/MPAL and 720 x 574 for PAL. However, the following restrictions apply to the left corner (XOrigin) and right corner (XOrigin + image width) of the image displayed on the TV.

$$XOrigin \leq 307 \text{ (NTSC/MPAL), } 300 \text{ (PAL)}$$

$$XOrigin + \text{image width} \geq 307 \text{ (NTSC/MPAL), } 300 \text{ (PAL)}$$

The non-display region generally includes the synchronization signal and signals that differ from graphics data (usually text broadcast signals and test signals). Developers are not able to display graphics in this region.

### 3. Safe Frames

Consider the "safe frame" to be the display region inside which information that is essential to the game can be safely drawn to almost any TV screen.

The safe frame for the Nintendo GameCube™ will vary depending on the type of TV, but in general, it is a little under 640 pixels horizontally and around 400 pixels vertically. Nintendo GameCube™ pixels are taller than they are wide, so images can be displayed to almost 640 pixels in the central horizontal position of the TV screen. However, many TVs have rounded corners, and in such cases, the images will not be completely displayed. There is also an area of about ten pixels at the top and bottom of most TV screens where content cannot be displayed. As a result, you need to be careful about the four corners as well as the top and bottom of the screen when displaying important information.

Figure 2 shows some examples of safe frames. Please consult these examples and consider the importance of information when deciding where to position the display of that information.

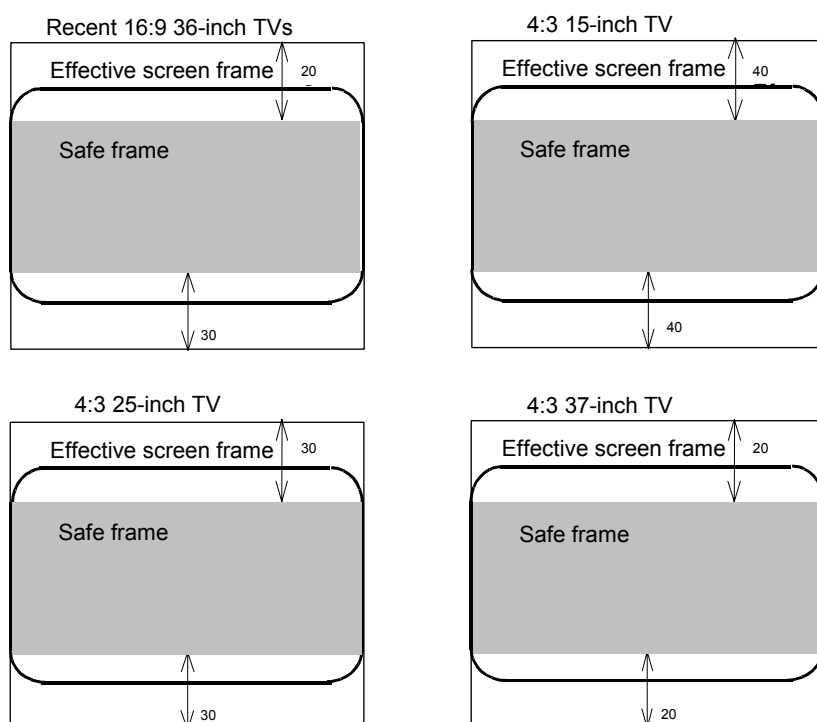


Figure 2: Safe frame examples (effective screen frame = 640 x 480)

### 4. Pixel Ratios

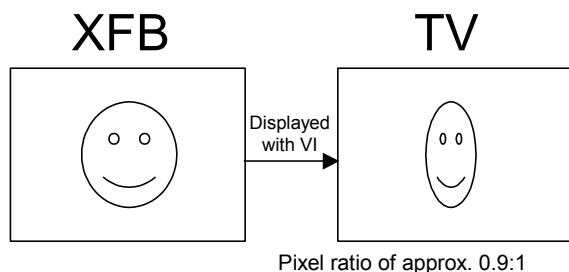
With the Nintendo GameCube™, when pictures that have been written to the Embedded Frame Buffer (EFB) are copied to the External Frame Buffer (XFB, the frame buffer stored in image memory), you can stretch the images in the vertical direction.

When graphics data in the XFB are displayed on the TV screen, you can use the scaling functions of the Video Interface (VI) to stretch the display screen horizontally. When the display is stretched by the VI in the horizontal direction, 8X oversampling is performed, followed by interpolation using a 49-step scaling filter and then resampling to regain the original dot clock. These scaling functions can be used to change the pixel ratio of the Nintendo GameCube™.

The sections below describe the pixel ratios when images are displayed on standard TVs without the use of scaling.

## 4.1 Displaying on a normal size (4:3) TV

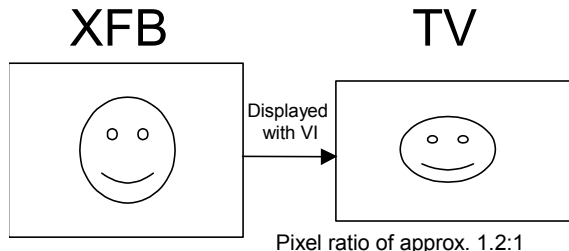
The pixel ratio is approximately 0.9:1.0. The pixel is shaped like a vertically elongated rectangle. Even if XFB contains square data, if VI scaling features are not used somewhat vertically elongated rectangles will be displayed on the television screen. If you want to display a 640 x 480 screen image with a 1:1 pixel ratio, stretch the 576 pixels inside the XFB to 640 pixels (since approximately 0.9:1.0 equals approximately 576:640). To read about how to use scaling in the horizontal direction, see the page on the VIFunction in the Nintendo GameCube™ Function Reference Manual.



**Figure 3: Pixel ratio when scaling is not used (for a 4:3 TV)**

## 4.2 Displaying on a wide-screen (16:9) TV

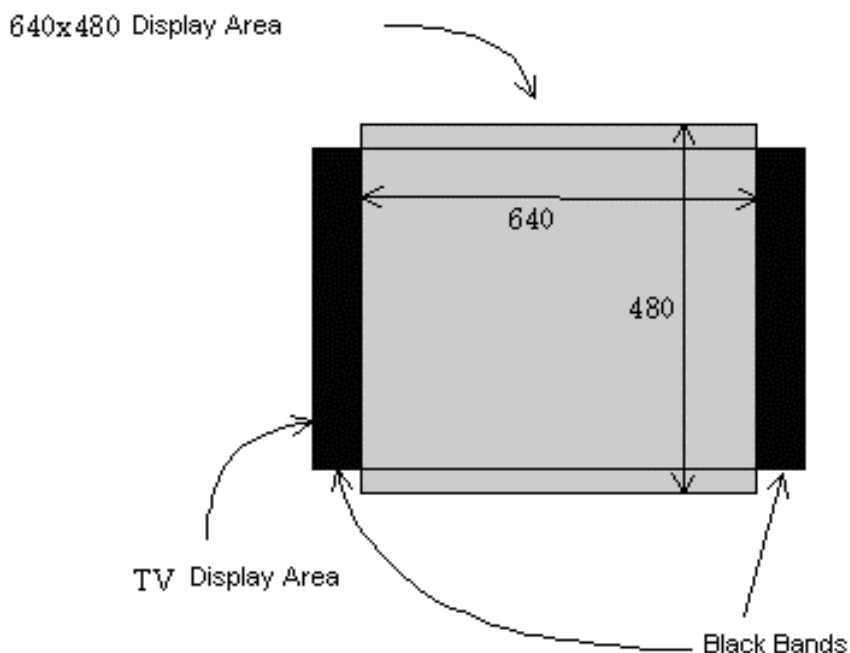
The pixel ratio is approximately 1.2:1.0. The pixel is shaped like a rectangle that is longer in the horizontal direction. If the VI scaling functions are not employed, then even data for a square shape in the XFB will be displayed on the TV screen as an oblong shape that is slightly longer in the horizontal direction. If you want to display with a 1:1 pixel ratio, then you need to place the image in the XFB as an oblong shape that is longer in the vertical direction. To read how to use scaling in the vertical direction when copying data from the EFB to the XFB, see the page on the GXSetDispCopyYScale function in the Nintendo GameCube™ Function Reference Manual.



**Figure 4: Pixel ratio when scaling is not used (for a 16:9 TV)**

## 5. Black Bands on Both Edges of the TV Screen

When the number of horizontal display pixels is 640, there may be times (depending upon the television) when there are vertical black bands that stand out on both sides of the non-display region. For such televisions, there are more than 640 pixels of display area. So if a display of only 640 pixels is selected, the outside edge will appear as black bands. When the number of vertical display pixels is 480, there will be an area of several tens of pixels above and below that is not displayed, as described in paragraph 4.1 "[Displaying on a normal size \(4:3\) TV](#)". For this reason, these black bands do not appear.



**Figure 5: Black bands on both edges of the screen (The black bands are drawn dark, but are not actually that thick.)**

We have noticed that these black bands "flicker" on some televisions. This happens on televisions where the bright pixels appear more oblong than dark pixels. On such televisions, where there are many lines of bright pixels, the black bands are thin. And in lines where there are many dark pixels, the black bands are thicker. In games that have lines that suddenly change the brightness, the width of the black bands will change over time and appear to be flickering.

The existence of these black bands themselves is not such a serious problem, but on televisions where there is significant flicker it could interfere with the progress of the game. Furthermore (and the detailed reasons are not understood), this problem is apparently especially widespread on televisions in Europe. (The developers have been asked whether or not they will solve this problem. If they do, the following methods are available, so please use them for reference.)

This problem can be improved by using the `VIXscale` function, to stretch the display area horizontally by around 10 ~ 30 pixels. Doing this reduces the width of the black bands, and makes any flicker less apparent. More specifically, the process uses the values shown below as the `VIWidth` of the structure `GXRenderModeObj`, which can then be passed to the function `VIConfigure`. Please correct the `VIXOrigin` at the same time, by centering the image on the screen.

Amount to Stretch to the Left and Right	5 Pixels	10 Pixels	15 Pixels
<code>VIWidth</code>	650	660	670
<code>VIXOrigin</code>	35	30	25
Width of Black Band	Thinner, but still there	Disappears on some televisions, remains on others	Disappears on nearly all televisions
Horizontal Safety Frame	It's better to allow about 5 pixels to the right and left	It's better to allow about 10 pixels to the right and left	It's better to allow about 15 pixels to the right and left

**Table 1: Example of screen width and safety frames taking into account the flicker of the black bands on both edges of the screen**

When `VIWidth` = 650, the black bands will continue to exist, but they can be made narrower. As a result, the black band flicker can be reduced somewhat. When `VIWidth` = 660, there will be televisions where the black bands disappear, as well as those where they remain. When `VIWidth` = 670, the black bands will disappear on nearly all televisions.

Please allow a margin equal to the respective number of pixels stretched, as a safety frame, when making this sort of horizontal stretch. In other words, if the right and left sides are stretched by 5 pixels each, then 5 pixels would be taken from the right and left edges. If the stretch were 10 pixels, then there would be an area equal to 10 pixels on each side in which no text or images that were important to the progress of the game would be displayed.

For reference, it has been decided to **stretch the display area 10 pixels, and** establish a 10-pixel safety frame, on both the right and left sides for both NTSC and PAL systems on all future games created by Nintendo.

Furthermore, when the number of vertical display pixels is less than 480, there could be (horizontal) black bands above and below the display. But because of the way the televisions are configured, the black bands will not flicker, so this will not be a problem. This will also not be a problem when trying to create a movie-theater effect by purposely blacking out the top and bottom in order to approximate a 16:9 aspect ratio.

Additionally, by stretching the entire screen horizontally, each pixel will be stretched horizontally by approximately 1.56% (for a 5-pixel stretch on both sides), approximately 3.12% (same: 10 pixels) and 4.69% (same: 15 pixels). Please refer to paragraphs 4.1 [“Displaying on a normal size \(4:3\) TV”](#) and 4.2 [“Displaying on a wide-screen \(16:9\) TV”](#) when considering pixel ratios.

## 6. MPAL

Application software that is destined for North America may also be marketed in Brazil. However, you do not need to include Portuguese-language support.

When the Nintendo GameCube™ has an MPAL IPL, the executing application will automatically display in MPAL video mode. However, if the application software uses customized VI settings ( see the Video Interface API section in the "Dolphin Reference Manual" ), then the application is responsible for manually changing these settings.



## 7. PAL

Detailed information on PAL can be found in the “*Nintendo GameCube™ PAL Game Development Guidelines*”.

### 7.1 Supporting Development Hardware

We plan to support PAL with various hardware in the future. For PAL support, all that will need to be done is to replace the video board in the development machine. Once PAL support is added, video boards will support all formats -- NTSC, MPAL and PAL. The functionality to switch between NTSC, MPAL, and PAL will be accomplished using a dip switch on the GDEV and via commands on the DDH.

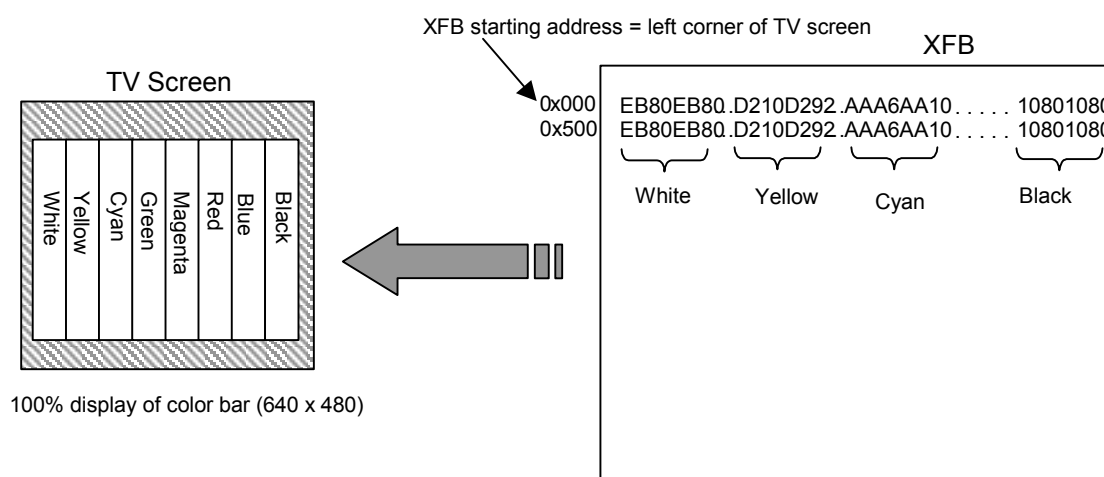
### 7.2 Handling EU RGB60 Mode

In EU RGB60 mode the field frequency is 60 Hz and the 480 active scan lines are output as an RGB signal. This mode allows you to produce a more vivid and smoother display than with PAL, which has a field frequency of 50 Hz and 574 active scan lines. For an application that is for a PAL target country, the required display format is 50 Hz. Thus, applications supporting EU RGB60 mode need to have display formats for both 50 Hz and 60 Hz. Support for EU RGB60 mode is not required for all games.

## 8. Handling YUV Data

The image data that will be displayed on the TV screen (the final image) must be placed in the XFB in YUV 4:2:2 format (YUYV...). Accordingly, images that you want to directly display to the XFB must be written in YUV format. The image data on the XFB corresponds to 2 pixels of YUYV data (with a size of 4 bytes). The XFB handles YUYV listed data as a single unit, so if you want to write data directly to XFB, you will need to write in 2-pixel increments.

Figure 6 shows the correspondence between the YUV data on the XFB and the TV screen. (To read in detail about the YUV format in the XFB, see "Section 12. Video Output" in the Graphics Library.)



**Figure 6: Correspondence between TV screen and YUV data in the XFB**

The YUV values are obtained using the following conversion equations:

$$Y = 0.257 * R + 0.504 * G + 0.098 * B + 16$$

$$U = -0.148 * R - 0.291 * G + 0.439 * B + 128$$

$$V = 0.439 * R - 0.368 * G - 0.071 * B + 128$$

You can calculate the YUV values by substituting actual numbers from 0 to 255 for the RGB variables in these equations, but the YUV values are not continuous quantities. The obtained YUV values fall within the ranges shown below, but some combinations within the ranges cannot actually be obtained.

Y : 16 ~ 235

U : 16 ~ 240

V : 16 ~ 240

Please do not write unobtainable combinations of YUV values to the XFB. For example, say you want to clear the screen by filling it with black. With RGB you would write (R,G,B) = (0,0,0), but with YUV you would write (Y,U,V) = (16,128,128). Data listed as [16,128,16,128...] would be written to the XFB. Images are normally written to the EFB in RGB format and then automatically converted to YUV format when they are copied from the EFB to the XFB, so there is no particular need to worry about YUV data. However, be careful about handling YUV data when writing values to the XFB.

## 9. Problems with Screen Noise After IPL Ends

Normally, the Nintendo GameCube™ EFB is cleared when the GXDispCopy function transfers data from the EFB to the XFB. When GXDispCopy is performed in the application software for the first time, the EFB has not been cleared yet and the data that was there is sent to the XFB. Accordingly, if you start video output with VSetBlack, this EFB data from before the clear will be displayed as noise on the screen. You can prevent this from happening by not outputting the data that is transferred by the GXDispCopy function the first time it executes, and instead waiting for the second output from this function to begin video output. Otherwise, you need to implement some measure to clear the EFB such as by creating an image that will fill the entire screen with polygons.

## 10. About Progressive Mode

Interlaced Mode is the splitting of one 1/30 second frame into two frames. In the first 1/60 second the odd numbered scan lines are displayed, and in the next 1/60 second the even numbered scan lines are displayed. Progressive Mode is the displaying of all scan lines in 1/60 second.

The Nintendo GameCube™ has the ability to display in Progressive Mode (480 enabled scan lines, D2 terminal specification). Usually, Progressive Mode can only be displayed on a television that supports Progressive Mode.

Interlaced/Progressive mode switching means that the Nintendo GameCube™ video output mode can be switched between Interlaced Mode and Progressive Mode. If the application is to support Progressive Mode this switching processing must be carried out, but if there is no support for Progressive Mode no switching processing is needed.

### 10.1 Switching between Interlaced Mode and Progressive Mode

The operations that perform switching between Interlaced Mode and Progressive Mode take place between the time when power is applied to the Nintendo GameCube™ and when the game application starts. The developer needs to execute a three-step check before the actual switch is done. (See the flowchart in Diagram 7.)

There is a progressive verification flag check in the three-step check. The progressive verification flag uses 1 bit of SRAM in the Nintendo GameCube™. It is referenced as one condition to display the message querying the user if they want to switch. The OSGetProgressiveMode function checks the flag information. The OSSetProgressiveMode function sets the flag to ON or OFF. The setting when shipped from the factory is OFF.

The actual procedure is as follows. First, check whether a D-terminal video cable or component video cable is plugged into the Digital AV Connector of Nintendo GameCube™. Use the VGetDTVStatus function to capture this information.

Next, check whether the B Button is being pressed when the game program starts, or whether the progressive flag is set to ON. Although the user may be pressing the B Button from the point when the Nintendo GameCube™ logo was displayed, the B Button's status can only be determined after the logo has been displayed but before the game's initialization screen (the first screen displayed on the TV screen by the application) is displayed. Finally, if either one of the following conditions applies, display a message, "Do you want to display the game in Progressive Mode? Yes or No?" to the player.

(Please note that even applications that are progressive-compatible will display in interlaced mode every time that the Nintendo GameCube™ is started.)

- A cable is plugged into the Digital AV Connector, and the B Button is pressed while the game operations begin.
- A cable is plugged into the Digital AV Connector, and the progressive flag is set to ON.

If the progressive flag is set to ON (1), it means that if there is a cable plugged into the Digital AV Connector, the user query message will be automatically displayed each time the game starts.

But if the progressive flag is set to OFF (0), it means that if there is a cable plugged into the Digital AV Connector and the user must press the B Button when the game starts up, in order to bring up the query message.

The default cursor position should be set to "Yes" for the user query message. If the user selects "Yes," then the progressive flag should be set to ON, and the game should be displayed in Progressive Mode from this point forward. If the user selects "No," then the progressive flag should be set to OFF, and the game should be displayed in Interlaced Mode from this point forward.

Use the VConfigure function to actually switch the display format from interlaced to progressive mode. Please refer to Chapter 11 "[Switching the Render Mode](#)" for more detail. Please note that turning the progressive confirmation flag ON by itself does not mean that the display has switched to Progressive Mode.

In cases where it requires a long time for data to load when the game starts, a user inquiry message is displayed while waiting for the game title to display. The timer is set for this message so it displays for approximately 10 seconds. If a timeout occurs before the user chooses a selection, processing is carried out automatically along the branch corresponding to the current location of the cursor. During this time it is recommended that the loading routines for the various types of data be carried out in the background.

In cases where it requires only a short period of time for data to load when the game starts, the user inquiry message is displayed at the same time as the game title. Until the user selects a mode, make sure the mode does not switch to progressive automatically. Use the most appropriate user inquiry message based on the application software in use.

## 10.2 Requirements when Using the Progressive Mode

In order to make it easier for your users to select Progressive Mode, when you ask them, "Do you want to display the game in Progressive Mode?", make the default position of the cursor "Yes."

In the Instruction Booklet for applications that support Progressive Mode, include a brief description of Progressive Mode and a note saying something like, "In order to activate Progressive Mode, start pressing the B Button when the Nintendo GameCube™ logo is displayed (or while the logo is being displayed), until a message like, 'Do you want to display the game in progressive mode?' comes up. A cable must be plugged into the Digital AV Connector to enable Progressive Mode."

For some brands of televisions, input of a progressive signal will set and fix the TV to a 16:9 display mode. Include a caution to this effect in the Instruction Booklet for applications that support Progressive Mode.

Nintendo GameCube does not support high definition output, so do not use any descriptions containing HDTV.

Applications for PAL target countries do not provide support for progressive scan mode, so completely remove any progressive scan support routines when creating PAL applications.

## 10.3 Recommendations when Using Progressive Mode

### 10.3.1 Time to complete the switch

The TV screen will be distorted the instant the display switches from Interlaced Mode to Progressive Mode. The time it takes the screen to return to normal (the “switch completion period”) differs from TV set to TV set, but generally several dozen frames are required (see Table 1). Since the exact performance of the television cannot be predicted during the switch completion period, do not display important messages or scenes immediately afterward. After the switch is complete, you could display a message screen that waits for user input before proceeding, such as “The display mode has switched to Progressive Mode. Press the A Button to continue.”

TV Type	Production Date	Number of Frames Required to Complete the Switch
36" (Company A)	Jul-Dec, 1998	80 or 140 frames
28" (Company A)	Jul-Oct, 2000	40 frames
36" (Company B)	Jul-Oct, 2000	90 frames

**Table 2: Market Survey of Differences between Televisions that Support Progressive Mode**

In order to control the distortion of the TV screen, after querying the user about whether or not to switch to Progressive Mode, use the following sequence to complete the switch to Progressive Mode.

Fade the interlaced screen displaying the Progressive Mode query to black.

1. Display a black screen.
2. Switch display modes.
3. Display a black screen.
4. Wait for several dozen frames.
5. Fade in a screen that does not contain an important scene or message.

### 10.3.2 Processing Hardware Resets during Progressive Output

For some brands of televisions, input of a progressive signal will set and fix the TV to a 16:9 display mode. If the user switches the television screen mode, a 4:3 display may be possible. However, depending on the television, once the progressive signal is cut off it may return to 16:9. When displaying in Progressive Mode, we recommend that you do not return to Interlaced Mode if the RESET Button is pressed or if there is a significant change in process flow. For more details on reset processing, refer to the “Nintendo GameCube Reset Guidelines”.

### 10.3.3 Other considerations for Progressive Mode

Switching the display mode takes place when starting up the game for the first time, and this configuration information is saved as the progressive flag to the Nintendo GameCube™ itself. Please limit display mode switching to this one time. For example, do not add operations to your game that switch display modes in response to detection of D-terminal video cables or component video cables that have been plugged in or removed during gameplay.

## 10.4 Interlaced/Progressive Mode Switching Flowchart

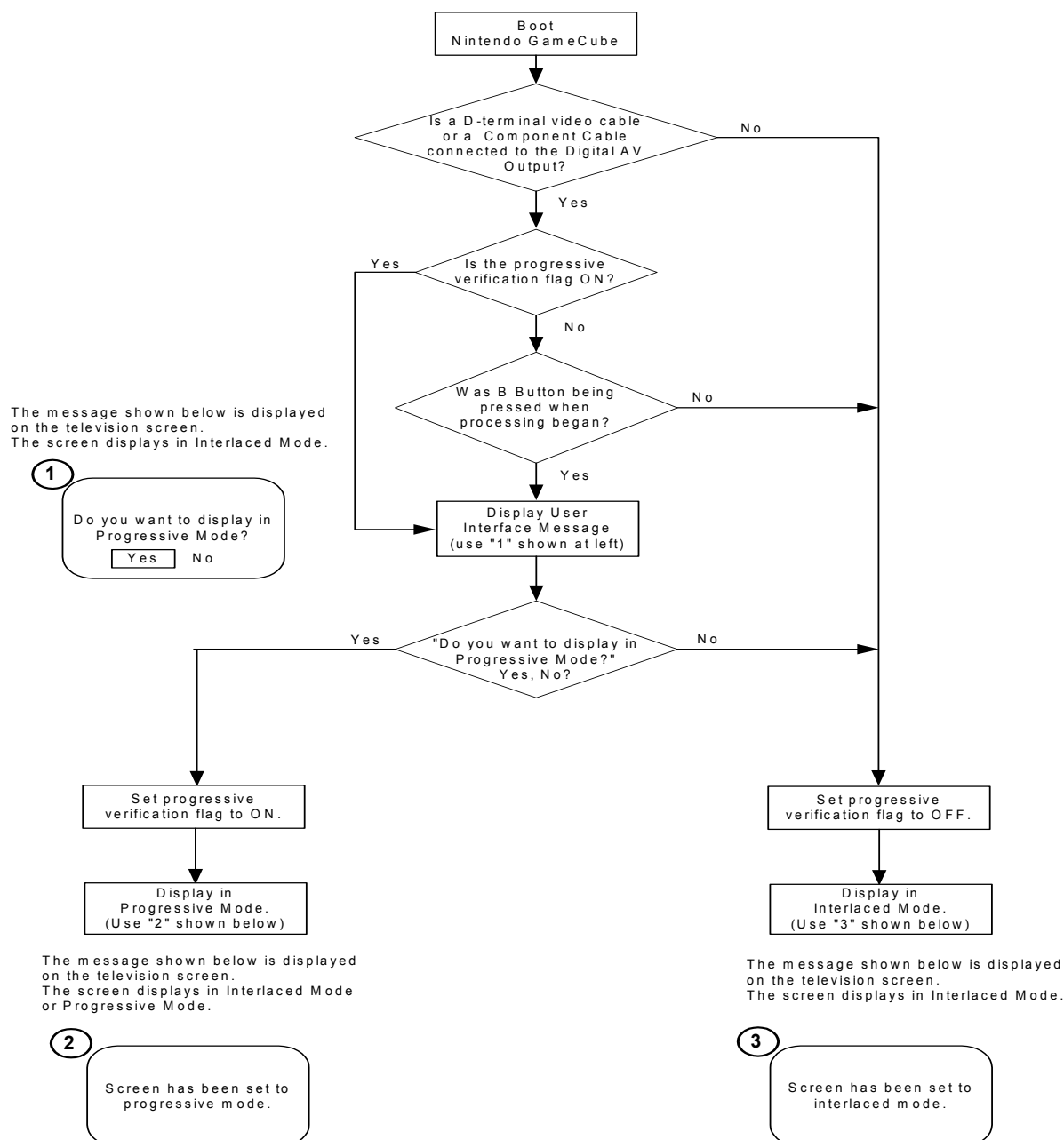


Figure 7: Flowchart Illustrating Switching between Interlaced and Progressive Modes

日本語メッセージ	English message
プログレッシブモードで表示しますか？	Would you like to display in progressive scan mode?
はい	Yes
いいえ	No
画面表示モードはプログレッシブモードにセットされました。	Screen display has been set to progressive scan mode.
画面表示モードはインターレースモードにセットされました。	Screen display has been set to interlace mode.

Table 3: Sample of messages displayed on screen during switch between Interlaced and Progressive Modes

## 11. Switching the Render Mode

When using the `VIConfigure` function to do the following switching, call `VIWaitForRetrace` twice as shown in Code 1 below. This is the same for NTSC, PAL, and MPAL.

Switch from:

- Interlace to Non-interlace
- Interlace to Progressive
- Non-interlace to Interlace
- Non-interlace to Progressive
- Progressive to Interlace
- Progressive to Non-interlace

---

```
rmDS = &GXNtsc240Ds;
rmINT = &GXNtsc480Int;
ds = 1;

while(1)
{
    ... ..

    if (ds)
        VIConfigure(rmDS);
    else
        VIConfigure(rmINT);

    // Wait for 2 fields when changing video mode
    VIFlush();
    VIWaitForRetrace();
    VIWaitForRetrace();

    ds ^= 1;
}
```

---

**Code 1: Switching the Render Mode between NTSC Interlace and NTSC Non-interlace**

## 12. About Televisions that Support Progressive Mode

Depending on the television that supports Progressive Mode, the video signal (interlaced, non-interlaced, low resolution non-interlaced) the television receives may have been scan line interpolated before being displayed. The user may be using such a television, so the following points need to be considered.

- If low-resolution Non-interlaced Mode (Double-Strike Mode) display is used on a television that supports Progressive Mode, the same line can be displayed twice, creating an unnatural, jagged effect. This issue also occurs when displaying low-resolution non-interlaced images from the N64 or other game console on a television that supports Progressive Mode. You may be able to display important messages in small type, or detailed images, on a television that does not support Progressive Mode. But the result may not be as good as with a television that supports Progressive Mode. We recommend that a high resolution mode be used in these cases.
- As a special effect, some games insert white screens between each frame in order to achieve a flashing effect. If you use such an effect in your game, be aware that the television set's linear interpolation can prevent the desired flashing effect from being displayed. To achieve a flashing effect, insert a white screen every other frame. If the flash time is short, a sufficient effect may be achieved if you insert a white screen every frame. Therefore, verify this effect with a television which supports Progressive Mode and determine what level of white screen insertion is required.

## 13. Other Items

Visually-induced seizures, seizures precipitated by flashing lights and patterns, may occur in some individuals who have a pre-existing susceptibility called photo- or pattern sensitivity. While photo- and pattern sensitivity in the general population are extremely rare, persons with either of these conditions may experience seizures as a result of watching some kinds of television pictures or playing certain video games. Therefore, we suggest avoiding excess flickering or flashing.

The Nintendo GameCube™ is designed on the assumption that display is to a 4:3 screen. Please use whichever method you deem optimal to add support for 16:9 screen TVs to your application software.

